JOURNEY TO RESILIENCE

Vilas Veeraraghavan, Director of Engineering, Walmart

CHAOS ENGINEERING

We know what it is.

270M

11K+

500B+

WHERE DO WE START THE JOURNEY?

TRUTHS

- Reliability is no longer a function of redundancy and over-scaled hardware.
  - Specifically, to exist in a hybrid cloud environment, we have to acknowledge that cloud providers are an external dependency which are reliability risks.
- Customers expect more and ‘scheduled downtime’ is no longer an acceptable term.
  - A user performing transactions (search, add to cart, payment) should not perceive a loss of functionality due to systemic failure.
- Users can lose trust on brand due to a single bad experience
  - Loss could be temporary OR lifetime

GOAL

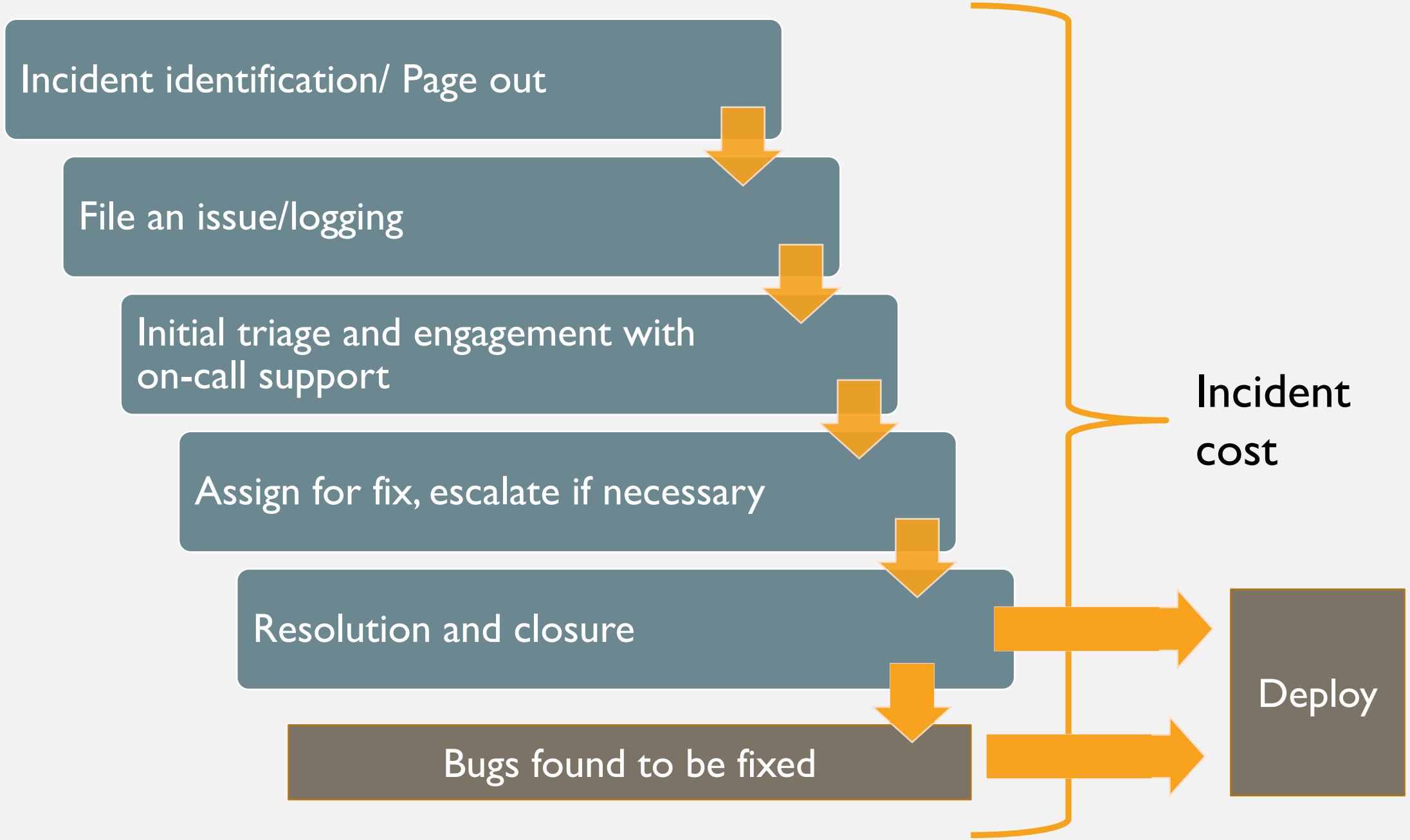
To maintain an application ecosystem where failures in infrastructure and dependencies cause minimal disruption to the end user experience

EVERY OUTAGE IS A CHAOS EXERCISE

- Just not intentional
- Teams discover gaps and fix it
- Revenue impact can be huge

DOWNTIME IS EXPENSIVE

- Calculate incident costs
- Break it down by quarter to find trends
- Track it and plan better to establish a culture of software resilience


Incident identification/ Page out
File an issue/logging
Initial triage and engagement with on-call support
Assign for fix, escalate if necessary
Resolution and closure
Bugs found to be fixed
Incident cost
Deploy

THE HOMEWORK

OBSERVABILITY

- Logging alone is not enough
- Ability to differentiate between healthy and unhealthy behavior
- Alerts set up
- Make sure your on-call knows what to do for specific problems
- Can an on-call engineer narrow down an issue successfully within the SLA of a P1 ?

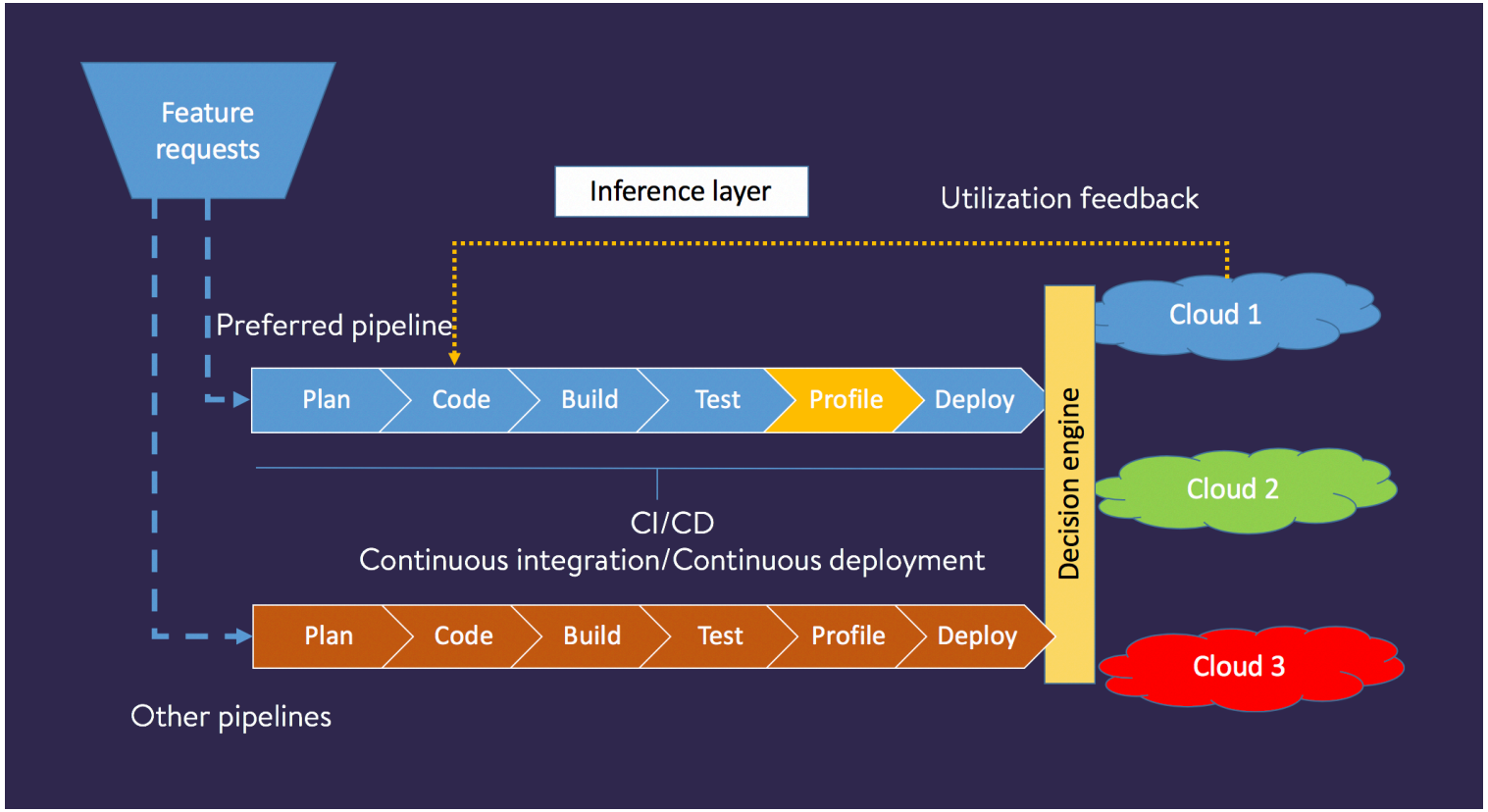
MORE ON-CALL PREREQUISITES

- Create Disaster Recovery (DR) failover playbook
- Define critical dependencies
- Compose playbook for critical dependency failures
- Define non-critical dependencies
- Define thresholds at which non-critical dependency failures will impact system

GENERATING PRODUCTION-LIKE LOAD

- Load generation that can replicate production
- No point of running a “failure injection” test if you cannot verify and characterize behavior
- Build OR buy – no preference unless proprietary reasons

CI/CD WORKFLOW – INVEST IN IT



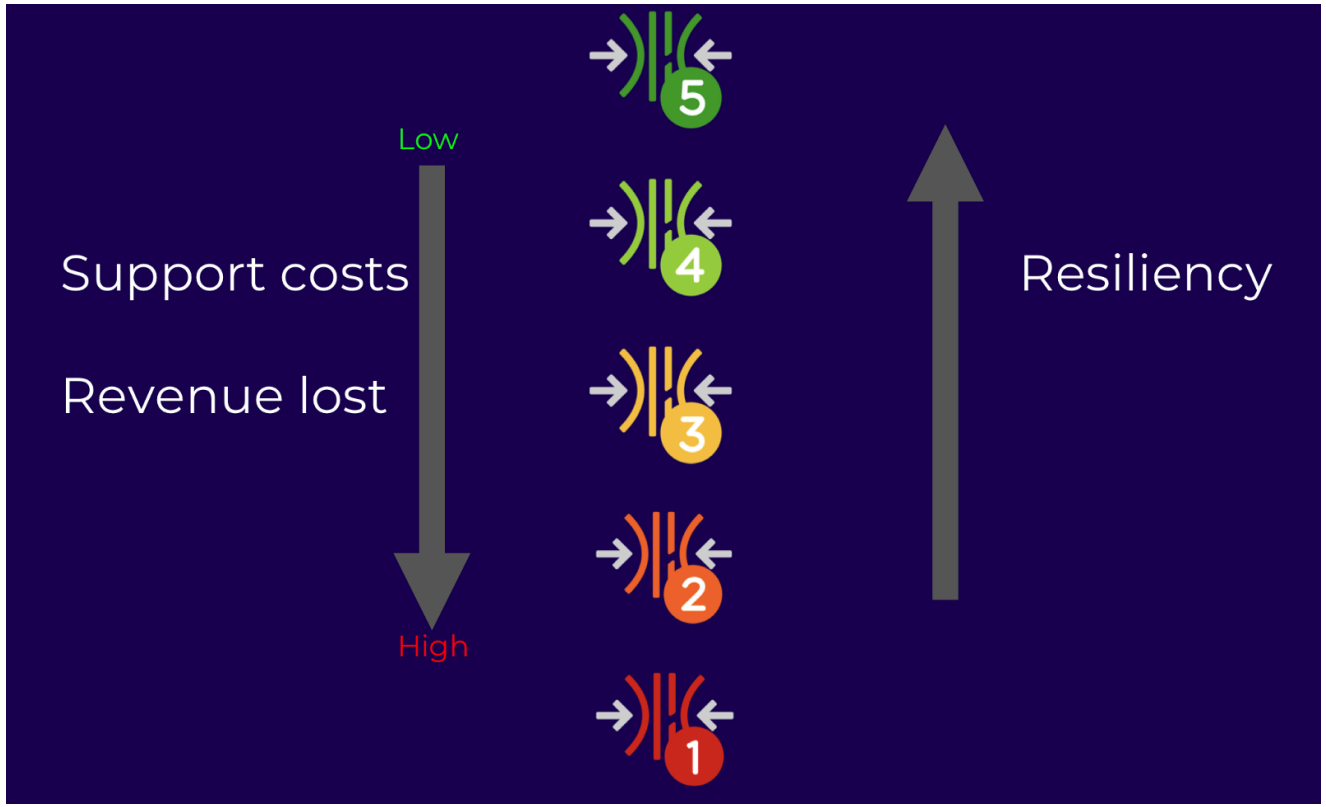
BUILD A MATURITY MODEL

Charting a path to software resiliency

Vilas Veeraraghavan

Oct 2, 2018 · 7 min read

SUPPORT COSTS



BUILD THE RIGHT TOOLS

Resiliency Doctor — A tool to achieve resiliency in hybrid cloud application ecosystems

Vijitha S Murthy Follow
Jul 18, 2019 · 6 min read

By Vilas Veeraraghavan, Vijitha S Murthy

BUILD THE RIGHT MINDSET

- Trainings
- Certifications
- Brown bag sessions – for case studies
- Build resilience after outages with blameless post mortems
- Carrots, not sticks

WHAT WE LEARNT ON OUR JOURNEY

ELIMINATE VANITY POSITIONS

- ”Chaos Practitioners” or “Resiliency experts”
- Every engineer needs to feel empowered to run exercises
- Exercises cannot be conducted without complete participation

DON'T ASSUME. VERIFY.

- Observability – teams are not as well-instrumented as they think they are
- On-call is being reactive. Test them to be sure
- Ask teams what their goals are with Resiliency
- Verify their deployment pipelines

ARE TEAMS READY FOR EXERCISES?

- They never are. Initially
- Build solid training
- Allow them to fail. But not in prod

WHERE ARE WE NOW?

REPORT CARD

- Application teams eager to run exercises
- Management confidence in the program
- Increased resilience in engineering team to deal with outages and to conduct subsequent learnings
- Application owners have better understanding of degraded state expectations

THANK YOU